Pacific Graphics 2011
Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin
(Guest Editors)

*Volume 30 (2011), Number 7*

# Toward Optimal Space Partitioning for Unbiased, Adaptive Free Path Sampling of Inhomogeneous Participating Media

Yonghao Yue[1], Kei Iwasaki[2], Bing-Yu Chen[3], Yoshinori Dobashi[4] and Tomoyuki Nishita[1]

[1]The University of Tokyo [2]Wakayama University [3]National Taiwan University [4]Hokkaido University
[1]{yonghao,nis}@nis-lab.is.s.u-tokyo.ac.jp [2]iwasaki@sys.wakayama-u.ac.jp [3]robin@ntu.edu.tw [4]doba@ime.ist.hokudai.ac.jp

**Abstract**

*Photo-realistic rendering of inhomogeneous participating media with light scattering in consideration is important in computer graphics, and is typically computed using Monte Carlo based methods. The key technique in such methods is the free path sampling, which is used for determining the distance (free path) between successive scattering events. Recently, it has been shown that efficient and unbiased free path sampling methods can be constructed based on Woodcock tracking. The key concept for improving the efficiency is to utilize space partitioning (e.g., kd-tree or uniform grid), and a better space partitioning scheme is important for better sampling efficiency. Thus, an estimation framework for investigating the gain in sampling efficiency is important for determining how to partition the space. However, currently, there is no estimation framework that works in 3D space. In this paper, we propose a new estimation framework to overcome this problem. Using our framework, we can analytically estimate the sampling efficiency for any typical partitioned space. Conversely, we can also use this estimation framework for determining the optimal space partitioning. As an application, we show that new space partitioning schemes can be constructed using our estimation framework. Moreover, we show that the differences in the performances using different schemes can be predicted fairly well using our estimation framework.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—I.3.3 [Computer Graphics]: Picture/Image Generation—G.3 [Probability and Statistics]: Probabilistic Algorithms—

## 1. Introduction

Steam, water, fire, smoke, explosions, volcanic eruptions, clouds, atmosphere, mist due to waterfalls and splashes due to ocean waves are common participating media around us. These participating media are usually inhomogeneous. Photo-realistic rendering of these inhomogeneous participating media is important in computer graphics, as the rendered results are usually impressive. Indeed, the demand for high-quality rendering of those media is dramatically increasing in, *e.g.*, film industry.

Light scattering is essential in participating media rendering. Typically, Monte Carlo based methods [LW96, JC98, PKK00, RSK06] are used for solving the radiative transport equation [Cha50] to account for the light transport in participating media. A Monte Carlo based method generates a number of light paths to simulate the light transfer in the scene. The technique at the heart of such a method is the *free path sampling*, which is used for determining the distance (free path) between successive scattering events. The computation efficiency of participating media rendering and the quality of the resulting images highly depend on the free path sampling technique used.

In the computer graphics field, ray marching is generally used as the technique for free path sampling. However, as figured out by previous works, *e.g.*, Raab et al. [RSK06], ray marching results in a stochastically biased solution which does not converge to the exact solution. This bias depends on the sampling interval used in ray marching, and different sampling intervals result in different pixel colors.

On the other hand, an unbiased free path sampling technique called *Woodcock tracking* [WMHL65] was proposed in the nuclear science field. However, Woodcock tracking is known to become less efficient for more inhomogeneous participating media [Lep07]. To overcome this problem, re-

cently, unbiased acceleration techniques based on space partitioning were introduced. Yue et al. [YIC*10] used a kd-tree, and Szirmay-Kalos et al. [SKTM11] used a uniform grid. The key concept of these methods is to partition the spatial domain according to the spatial variation of the mean free path of the participating medium and to sample the free path in an adaptive manner.

In these unbiased, adaptive free path sampling techniques, the structure of the partitioned space influences the sampling efficiency, and an estimation framework for investigating the gain in sampling efficiency is important in order to obtain a good structure. However, currently, there is no estimation framework that works in the 3D space. For example, in [YIC*10], the space partitioning scheme they used is based on a heuristic approach that approximately solves the 3D problem by a set of disjoint 1D problems. As a result, the resulting sampling efficiency is not always optimal.

In this paper, we show for the first time an estimation framework that works in 3D space. Using our new framework, we can analytically estimate the sampling efficiency given a particular space partitioning. Conversely, we can also use this estimation framework for determining the optimal space partitioning. Our framework is derived from the formulation of the 1D problem presented in [YIC*10], and can be regarded as a generalization of it. Thanks to the generality, our estimation framework is the first one that can be used to develop automatic space partitioning schemes based on any typical spatial subdivision structures. In this paper, we show new automatic space partitioning schemes aiming at optimal sampling efficiency, using a uniform grid, octree and kd-tree. Moreover, we show that the differences in the performances using different schemes can be predicted fairly well using our estimation framework.

## 2. Related Work

In this section, we focus on previous work related to free path sampling. For previous research on participating media rendering, the reader may refer to the surveys [CPP*05, GJJD09].

The techniques for free path sampling can be classified into the following two categories: 1) ray marching or its variants [PH89, JC98, PKK00, BM03], and 2) Woodcock tracking or its variants [WMHL65, CCW72, RSK06, Lep07, YIC*10, SKTM11].

Although ray marching is widely used in the computer graphics field, it produces a biased solution which does not converge to the exact solution. The bias due to ray marching could be problematic because 1) different sampling intervals would result in different results and 2) the error between a biased solution and the exact solution is easily noticeable but is difficult to be estimated in advance.

An alternative that is free from the aforementioned bias induced problems is Woodcock tracking [WMHL65], which was proposed in the nuclear science field and was first introduced to the computer graphics field by Raab et al. [RSK06]. Woodcock tracking is proven to be unbiased by Coleman [Col68], thus will produce exact solutions. A shortcoming of Woodcock tracking is that it becomes less efficient for more inhomogeneous participating media [Lep07]. Badal and Badano [BB09] accelerated Woodcock tracking by implementing it on the GPU. Another improvement was done by Leppänen [Lep07], in which a two-level space partitioning was used. Very recently, much more efficient techniques using more general space partitioning schemes were proposed. Yue et al. [YIC*10] and Szirmay-Kalos et al. [SKTM11] proposed to use a kd-tree and a uniform grid, respectively.

In these space partitioning based methods, it is important to obtain a good structure of the partitioned space as the space partitioning influences the sampling efficiency. Moreover, we claim that an estimation technique to investigate the gain in sampling efficiency is important for determining how to partition the space. However, an estimation framework that works in 3D space is lacking. For example, when using uniform grids, the user needs to determine the resolution of the grid manually. The kd-tree based method offers an automatic space partitioning, but is inherently based on a heuristic approach that utilizes 1D formulations [YIC*10]. As a result, the method is imperfect, and the partitioned space is not guaranteed to be optimal. Thus, in this paper, we focus on deriving an estimation framework that works in 3D space, aiming at optimal space partitioning.

## 3. Free Path Sampling

In this section, we first state the free path sampling problem and briefly review Woodcock tracking and the concept behind previous methods utilizing space partitioning.

To generate a light path in a Monte Carlo based rendering method, scattering events are generated successively. To generate the $(i+1)$-th scattering event, the free path $d_i$ and the scattering direction $\vec{\omega}_i$ need to be determined through stochastic sampling. The scattering direction can be obtained using a conventional importance sampling technique. For the free path, we need to sample $d_i$ according to the following probability density function [PKK00]:

$$pdf_{fp}(\mathbf{x}_{i+1} = \mathbf{x}_i + d_i\vec{\omega}_i) = e^{-\tau(\mathbf{x}_i,\mathbf{x}_{i+1})}k(\mathbf{x}_{i+1}), \quad (1)$$

where $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are the locations of the scattering events $i$ and $(i+1)$, $\tau(\mathbf{x}_i,\mathbf{x}_{i+1}) = \int_{\mathbf{x}_i}^{\mathbf{x}_{i+1}} k(\mathbf{x}')d\mathbf{x}'$ is the optical depth between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, and $k$ is the extinction coefficient.

Woodcock tracking samples the free path by employing a rejection sampling technique. First, a *majorant* extinction coefficient $k_M$ which is never less than the extinction coefficient of the participating medium is computed. Then, Woodcock tracking samples pseudo scattering events by regarding the medium as a uniform medium with the extinction

coefficient being $k_M$. For unbiasedness, such pseudo scattering events are only accepted as 'real' scattering events with the probability $k(\mathbf{x}_i + d_i\vec{\omega}_i)/k_M$. When generating a new pseudo scattering event, the free path $d_i$ is incremented by $-\ln(1-rand())/k_M$, whose expectation value is $1/k_M$. The reader may also refer to [RSK06] for implementing Woodcock tracking.

Woodcock tracking becomes less efficient in more inhomogeneous participating medium for the following reasons. In an inhomogeneous participating medium, $k(\mathbf{x}_i + d_i\vec{\omega}_i)$ is often much smaller than the majorant extinction coefficient $k_M$, and the ratio $k(\mathbf{x}_i + d_i\vec{\omega}_i)/k_M$ becomes small where the medium is sparse. Therefore, pseudo scattering generation will be iterated many times until a real scattering event is detected.

Previous methods utilizing space partitioning improved Woodcock tracking by subdividing the spatial domain into multiple domains and used different majorant extinction coefficients for different partitioned domains. Unbiasedness is ensured as follows: when going across a partition, the sampling location is rewound back to the intersection point between the ray and the partition. Note that this rewinding operation requires an additional iteration.

The way toward finding the optimal space partitioning (*i.e.*, the space partitioning that makes the sampling efficiency optimal) is to minimize the number of iterations needed to generate a 'real' scattering event, taking into account the trade-off between the following two aspects: 1) if we partition the space more finely, the ratio of $k(\mathbf{x}_i + d_i\vec{\omega}_i)/k_M$ in each partitioned domain would be closer to its upper bound (*i.e.*, 1), and the number of iterations inside each partition would be decreased; 2) the number of the partitions should be kept small as we need additional iterations to go across these partitions due to the rewind process.

To account for the above trade-off, we need an estimation framework of the number of iterations, given a space partitioning. Currently, the only analytic estimation framework is the one formulated in 1D space [YIC*10]. In this paper, we present a generalized version of their estimation framework. Our framework is fully formulated in 3D space.

## 4. Evaluating the Partitioned Space

In this section, we first show an evaluation framework for 1D space, which is a slightly modified version of the formulation given by Yue et al. [YIC*10]. Then, we generalize it to 3D space.

### 4.1. Estimation Framework in 1D Space

The efficiency of the free path sampling is tightly coupled with the average numbers of the iterations needed before the rays encounter 'real' scattering events. Therefore, it is important to estimate the expected number of iterations.

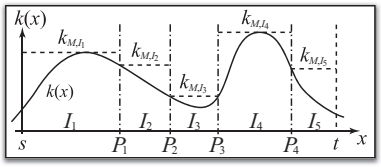



**Figure 1:** *Example distributions of a participating medium in 1D. The horizontal axis shows the locations x, and the vertical axis shows the value of $k(x)$.*

In 1D space (assuming the $x$ axis), the rays travel only along the $x$ axis, as shown in Figure 1. Suppose that we want to estimate the expected number of iterations when a ray travels through the interval $(s,t]$, and $k(x)$ is given at an arbitrary location $x$. For simplicity, we assume that a 'real' scattering event does not happen in the interval $(s,t]$.

For the interval $(s,t]$, we assume that we are given a space partitioning, which is represented by a set of $n$ subintervals, $I_j$, where $j = 1,...,n$, and the majorant extinction coefficient in a subinterval $I_j$ is given as $k_{M,I_j}$. Between any two adjacent subintervals $I_j$ and $I_{j+1}$, there is a partition $P_j$. When the ray travels through the interval $(s,t]$, the ray will pass through the subintervals and the partitions in alternate order. To estimate the expected number of iterations, we account for the expected numbers of iterations needed for passing through each subinterval and partition as follows.

Let $|I_j|$ be the length of the subinterval $I_j$. Since the majorant extinction coefficient in this subinterval is $k_{M,I_j}$, the expected distance that the ray will proceed in a single iteration is $1/k_{M,I_j}$. Thus, the expected number of iterations to travel through this subinterval is given by $|I_j|k_{M,I_j}$. To go across a partition, we need to perform a single rewind process. Thus, the expected number of iterations to go across a partition is 1. Combining these two results, the expected number of iterations $\bar{N}$ needed to travel the interval $(s,t]$ can be formulated as

$$\bar{N} = \sum_{j=1}^{n} |I_j|k_{M,I_j} + (n-1). \tag{2}$$

Similarly, we may also account for the expected computation time. Let $t_{iter}$ and $t_{rewind}$ be the average computation time for a single iteration inside a subinterval and for a rewinding process, respectively. Then, the expected computation time $\bar{T}$ for the ray to travel the interval $(s,t]$ can be formulated as

$$\bar{T} = t_{iter}\sum_{j=1}^{n} |I_j|k_{M,I_j} + t_{rewind}(n-1). \tag{3}$$

### 4.2. Estimation Framework in 3D Space

In 3D space, we need to account for any rays traveling in arbitrary directions. Thus, the formulation in 3D space is much more difficult than that in 1D space.

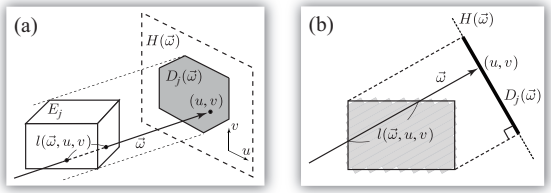



**Figure 2:** *(a): An example showing the space $E_j$ and its projection onto the direction $\vec{\omega}$. (b): Another view of (a) (the viewing ray is parallel to $H(\vec{\omega})$).*

To account for the rays in all the directions, we consider the average of the expected numbers of iterations for all the rays passing through a subspace or a partition. Since a typical space partitioning scheme uses axis-aligned planes, we assume the subspaces and the partitions are rectangular parallelepipeds and planes, respectively.

Let the bounding box of the participating medium be $E$. We assume that a space partitioning is given so that $E$ is subdivided into $n_s$ subspaces $E_j$ $(j = 1,...,n_s)$ by $n_p$ partitions $P_k$ $(k = 1,...,n_p)$. We estimate the average of the expected numbers of iterations $\bar{N}_{E_j}$ for the rays passing through a subspace $E_j$ as

$$\bar{N}_{E_j} = \int_{\Gamma(E_j)} k_{M,E_j} l_{E_j}(r)dr \Big/ \int_{\Gamma(E)} dr, \tag{4}$$

and the average of the expected numbers of iterations $\bar{N}_{P_k}$ for the rays passing through a partition $P_k$ as

$$\bar{N}_{P_k} = \int_{\Gamma(P_k)} 1 \cdot dr \Big/ \int_{\Gamma(E)} dr, \tag{5}$$

where $\Gamma(E_j)$, $\Gamma(E)$ and $\Gamma(P_k)$ indicate the sets of rays passing through $E_j$, $E$ and $P_k$, respectively. $k_{M,E_j}$ is the majorant extinction coefficient of the subspace $E_j$, $l_{E_j}(r)$ denotes the length of the intersection between the ray $r$ and the subspace $E_j$, and $dr$ is the measure of the ray $r$. The meanings of Eqs.(4) and (5) are as follows. The numerators in them describe the summation of the numbers of iterations for all the rays, and the denominators describe the size of the set of rays passing through $E$. Thus, by the division, we obtain the averages of the numbers of iterations for a single ray passing through $E$.

### 4.3. Analytic Solution

One of the important contributions of this paper is that the integrals appearing in Eqs.(4) and (5) can be calculated analytically by assuming 1) all the rays pass through the space $E$ and do not encounter 'real' scattering events and 2) the rays are distributed uniformly in the space. We show the analytical solutions below.

First, we show the analytical solution of $\int_{\Gamma(E_j)} k_{M,E_j} l_{E_j}(r)dr$. For simplicity, let us omit the subscription $E_j$ in $k_{M,E_j}$ and $l_{E_j}$. By calculating the integral first according to the directions of the rays, we obtain

$$\int_{\Gamma(E_j)} k_M l(r)dr = \int_{S^2} k_M \Big( \iint_{D_j(\vec{\omega})} l(\vec{\omega},u,v)dudv \Big) d\omega, \tag{6}$$

where $S^2$ denotes the set of all the directions, $d\omega$ is the measure of the solid angle for the direction $\vec{\omega}$. As shown in Figure 2(a), $D_j(\vec{\omega})$ is the projected region of the subspace $E_j$ onto a plane $H(\vec{\omega})$ which is perpendicular to the direction $\vec{\omega}$. $u$ and $v$ are the orthogonal coordinates in $H(\vec{\omega})$. $l(\omega,u,v) = l(r)$ is the length of the intersection between $E_j$ and the ray in the direction $\vec{\omega}$ passing through a point $(u,v)$ on $H(\vec{\omega})$. Let $|E_j|$ be the volume of $E_j$, then the following formula holds,

$$\iint_{D_j(\vec{\omega})} l(\vec{\omega},u,v)dudv = |E_j|, \tag{7}$$

because $l(\vec{\omega},u,v)dudv$ represents an infinitesimal pillar which is sliced from $E_j$ in the direction $\vec{\omega}$ as shown in Figure 2(b), and the integral of such pillars is identical to the volume of $E_j$. Therefore, we obtain

$$\int_{\Gamma(E_j)} k_M l(r)dr = 4\pi k_M |E_j|. \tag{8}$$

Next, we show the analytical solution of $\int_{\Gamma(E)} dr$. By calculating the integral first according to the directions of the rays, we obtain

$$\int_{\Gamma(E)} dr = \int_{S^2} \Big( \iint_{D(\vec{\omega})} dudv \Big) d\omega = \int_{S^2} |D(\vec{\omega})| d\omega, \tag{9}$$

where $|D(\vec{\omega})|$ denotes the area of $D(\vec{\omega})$, which is the projected region of the space $E$ onto a plane $H(\vec{\omega})$ perpendicular to the direction $\vec{\omega}$. $|D(\vec{\omega})|$ can be computed as follows. Let $\vec{e_x}$, $\vec{e_y}$ and $\vec{e_z}$ be the unit vectors along the $x$, $y$ and $z$ axes, respectively, and let $S_x$, $S_y$ and $S_z$ be the areas of the faces of $E$ perpendicular to the $x$, $y$ and $z$ axes, respectively. Then,

$$|D(\vec{\omega})| = S_x|\vec{e_x}\cdot\vec{\omega}| + S_y|\vec{e_y}\cdot\vec{\omega}| + S_z|\vec{e_z}\cdot\vec{\omega}|. \tag{10}$$

Let us use the polar coordinates $(\theta,\phi)$ to describe $\vec{\omega}$, *i.e.*, $\vec{\omega} = (\sin\theta\cos\phi, \cos\theta, \sin\theta\sin\phi)$. Then,

$$\begin{aligned}
&\int_{\Gamma(E)} dr = \int_{S^2} |D(\vec{\omega})| d\omega \\
&= 8\int_0^{\frac{\pi}{2}}\int_0^{\frac{\pi}{2}} (S_x \sin\theta\cos\phi + S_y\cos\theta + S_z\sin\theta\sin\phi)\sin\theta d\phi d\theta \\
&= 2\pi(S_x + S_y + S_z) = \pi S(E),
\end{aligned} \tag{11}$$

where $S(E)$ denotes the surface area of $E$.

Finally, we show the analytical solution of $\int_{\Gamma(P_k)} 1 \cdot dr$. By calculating the integral first according to the directions of the rays, we obtain

$$\int_{\Gamma(P_k)} 1 \cdot dr = \int_{S^2} \Big( \iint_{P_{k,\perp}(\vec{\omega})} dudv \Big) d\omega = \int_{S^2} |P_{k,\perp}(\vec{\omega})| d\omega, \tag{12}$$

where $P_{k,\perp}(\vec{\omega})$ represents the projected region of $P_k$ onto a plane perpendicular to $\vec{\omega}$, and $|P_{k,\perp}(\vec{\omega})|$ denotes its area. Then, using the polar coordinates for $\vec{\omega}$ to compute the integral, we obtain

$$\int_{\Gamma(P_k)} 1 \cdot dr = \int_{S^2} |P_{k,\perp}(\vec{\omega})| d\omega = 2\pi |P_k|, \tag{13}$$

where $|P_k|$ is the area of $P_k$.

Then, by putting Eqs.(4), (5), (8), (11) and (13) all together, we obtain an estimator for the average of the expected numbers of iterations $\bar{N}$ for a given space partitioning as

$$\bar{N} = \sum_{j=1}^{n_s} \bar{N}_{E_j} + \sum_{k=1}^{n_p} \bar{N}_{P_k} = \Big( \sum_{j=1}^{n_s} 4k_{M,E_j} |E_j| + \sum_{k=1}^{n_p} 2|P_k| \Big) \Big/ S(E), \tag{14}$$

where $n_s$ and $n_p$ are the numbers of the partitioned domains (subspaces) and partitions, respectively. Note that $\bar{N}$ is a dimensionless value because $k_{M,E_j}$ has the dimension $m^{-1}$, $|E_j|$ has the dimension $m^3$, and $|P_k|$ and $S(E)$ have the dimension $m^2$, respectively. The average computation time $\bar{T}$ can be estimated similarly as

$$\bar{T} = \Big( t_{iter} \cdot \sum_{j=1}^{n_s} 4k_{M,E_j} |E_j| + t_{rewind} \cdot \sum_{k=1}^{n_p} 2|P_k| \Big) \Big/ S(E). \tag{15}$$

An important contribution of the above formulations is that they do not contain any heuristic user specified parameters. They can be computed strictly.

## 5. On Optimal Space Partitioning Schemes

In this section, we apply our estimation framework to the development of new schemes aiming at optimal partitioning, using the uniform grid, octree and kd-tree, respectively.

### 5.1. Uniform Grid based Scheme

When using a uniform grid, we need to decide its resolution. That is, from a variety of the possible resolutions, we need to check for 1) the best resolution and 2) whether partitioning the space according to this resolution is beneficial.

First, we show that we can use Eq.(14) to estimate the average of the numbers of iterations. For simplicity, we assume that the simulation space (the bounding box) is a cube, and the length of each side is $w$. We also assume that each side is subdivided into $m$ slices so that we have $m \times m \times m$ grid-cells. Then, in Eq.(14), the volumes of the grid-cells are the same, thus $|E_j| = (w/m)^3$, the summation of the areas of the partitions is $\sum_{k=1}^{n_p} 2|P_k| = 3(m-1) \cdot 2w^2$, and $S(E) = 6w^2$. Thus, we have

$$\bar{N}_{part} = \frac{2w}{3} \cdot \frac{1}{m^3} \sum_{j=1}^{m^3} k_{M,E_j} + (m-1). \tag{16}$$

To find the best resolution, we find $m$ which makes $\bar{N}_{part}$ minimum from all possible $m$. Typically, we choose $m$ as a power of two.

To determine whether the partitioning according to this resolution is beneficial, we also estimate the average of expected numbers of iterations $\bar{N}_{nopart}$ without partitioning as,

$$\bar{N}_{nopart} = \frac{2w}{3} k_{M,E}. \tag{17}$$

If $\bar{N}_{part} < \bar{N}_{nopart}$, then it is beneficial to partition the simulation space according to the best resolution. Note that the above scheme can be straightforwardly extended to handle a simulation space represented by a rectangular parallelepiped or uneven number of slices for each side.

### 5.2. Octree based Scheme

We consider a partitioning scheme in a top-down and recursive style. We let the root node represent the simulation space (the boundary box). For the root node, first, we estimate whether it is beneficial to partition this node. If we decide to partition this node, then we recursively continue the partitioning for the eight child nodes.

For simplicity, again, we assume the simulation space is a cube, with the length of each side being $w$. We let $l$ be the level (or depth) of a node, *e.g.*, the root node is in the level 0. We partition a node by the axis-aligned planes containing the center of the node, so that all the eight child nodes have the same shape and size. Therefore, a node in the level $l$ is a cube, with the length of each side being $w/2^l$.

First, we estimate the average of the expected numbers of iterations $\bar{N}_{part}$ for the case with partitioning. Assume that the parent node is in the level $l$, then the volume of each child node is $(w/2^{l+1})^3$, the summation of the areas of the partitioning planes is $3(w/2^l)^2$, and the surface area of the simulation space $S(E)$ is $6w^2$. Thus, we have

$$\bar{N}_{part} = \frac{1}{4^l} \Big( \frac{2w}{3 \cdot 2^l} \cdot \frac{1}{8} \sum_{j=1}^{8} k_{M,E_j} + 1 \Big), \tag{18}$$

where $k_{M,E_j}$ denotes the majorant extinction coefficient for the eight child nodes. Next, we estimate the average for the case without partitioning as

$$\bar{N}_{nopart} = \frac{1}{4^l} \frac{2w}{3 \cdot 2^l} k_{M,E_p}, \tag{19}$$

where $k_{M,E_p}$ denotes the majorant extinction coefficient for the parent node. If $\bar{N}_{part} < \bar{N}_{nopart}$, then it is beneficial to partition the node.

Sometimes (*e.g.*, a participating medium with the high extinction coefficient portion coagulated around the center of a node), even if $\bar{N}_{part} < \bar{N}_{nopart}$ is not satisfied, the average of the numbers of iterations can be decreased if we continue partitioning. To handle such a case, we introduce the *fail count*, which is used when deciding the spatial subdivision structure for standard ray tracing.

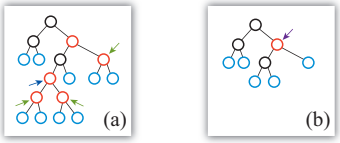



**Figure 3:** *(a) and (b): Before and after removing redundant partitioning. For simplicity, the number of child nodes is reduced to 2 instead of 8 in this illustration. Blue and red circles indicate the leaf nodes and the nodes with the fail count incremented, respectively. If the child nodes of a parent node (marked by the green arrow) are all leaf nodes and the fail count of the parent node is incremented, it is beneficial to merge the leaf nodes because $\bar{N}_{nopart} \geq \bar{N}_{part}$. The merge step can be processed recursively, e.g., for the node marked by the navy arrow. Conversely, if the child nodes of a parent node (marked by the purple arrow) are not all leaf nodes, it may be beneficial not to merge the child nodes, as there is benefit in the subsequent partitioning.*

A fail count is an integer value kept in each node during the partitioning process. Initially, the fail count of the root node is set to 0. During partitioning, if $\bar{N}_{part} < \bar{N}_{nopart}$ is not satisfied, we increment the fail count by 1 instead of terminating the partitioning process. The fail counts of the child nodes are set equal to that of the parent node. If the fail count of a node exceeds a threshold, we terminate partitioning.

After the octree is constructed, we try to remove redundant space partitioning (see Figure 3). That is, if the leaf nodes of a parent node are generated with the fail count increased, we can merge these leaf nodes. We perform this merge process recursively in a bottom-up fashion to remove all the redundant partitionings.

### 5.3. kd-Tree based Scheme

Our kd-tree based scheme is also in a top-down and recursive style. Given a space (let us call the space $E_p$), we check for each subspace in $E_p$ and estimate the benefit when partitioning the space using the planes containing the faces of the subspace. The average of the expected numbers of iterations with partitioning $\bar{N}_{part}$ can be estimated using Eq.(14). From all the possible subspaces, we find one that makes $\bar{N}_{part}$ minimum. Note that for a single subspace, there are many different ways for the partitioning. The summation of the areas of the partitions ($\sum_{k=1}^{n_p} 2|P_k|$) and the summation with respect to the majorant extinction coefficient ($\sum_{j=1}^{n_s} 4k_{M,E_j}|E_j|$) would differ according to the partitioning. So for a single subspace, we also check for all the possible partitioning. Then, we estimate the average of the expected numbers of iterations without partitioning $\bar{N}_{nopart}$. If $\bar{N}_{part} < \bar{N}_{nopart}$, we partition the space $E_p$ accordingly, and continue the partitioning process for the partitioned spaces recursively.

Although using this kd-tree based scheme will result in a fairly good sampling efficiency as demonstrated in Section 6, the partitioning scheme is currently time consuming because of the following two reasons. First, the number of possible subspaces is large. Assume that we use an auxiliary grid with $n$ grid-cells to search for the subspaces, then we have approximately $O(n^2)$ possible subspaces. Second, the number of possible partitioning for a single subspace is also large (*e.g.* if a subspace is fully contained in the space $E_p$, there are 426 different ways for partitioning. The number 426 can be obtained as follows. Distinct partitioning can be enumerated by considering how the space is subdivided by the partitioning planes. For each axis, there are two such planes, which we describe as $x^+$, $x^-$, $y^+$, $y^-$, $z^+$ and $z^-$. Then, a partitioning can be described as a permutation of these labels, like $y^+x^-x^+z^-y^-z^+$. The possible cases in the permutation, however, include duplicated cases, which can be omitted. That is, if two planes for the same axis, *e.g.*, $x^+$ and $x^-$, are neighboring, transpose the order of them would result in the same partitioning.)

Currently, we are limited to using an auxiliary grid which has equal to or less than $16^3$ grid-cells. This is because the computation time for space partitioning becomes expensive for higher resolutions. The computation times are approximately 2 seconds for $8^3$ grid-cells, 2.5 minutes for $16^3$ grid-cells, and 3 hours for $32^3$ grid-cells. We believe we can overcome this limitation in future research.

## 6. Evaluations

We evaluate the sampling efficiencies for the partitioned spaces obtained using the uniform grid based scheme (abbreviated as UG), the octree based scheme (OC), the kd-tree based scheme using the heuristic approach [YIC*10] (hKD) and the kd-tree based scheme presented in this paper (KD).

As we did not take into account the termination of iteration inside the analytical space in our estimation framework, the prediction of the numbers of iterations in actual situations is not accurate. Nevertheless, we show that our estimation framework can predict the ratio of the performances between any two schemes fairly well. Let us shortly explain the reason. For any straight line in the analytical space, there are a collection of random rays closely aligned with this line. The random rays can be assumed to be uniformly distributed in the collection, and we can connect some of these rays to generate a collection of rays that pass through the analytical space without termination of iteration inside the space. The average number of iterations for these connected rays can be assumed to be very close to that of the rays considered in our estimation framework. Thus, the average number of iterations for the random rays in the whole analytical space can be assumed to be the product of a constant and the estimated average number of iterations using our framework. When computing the acceleration ratio between any two schemes, such constants will be canceled out.

The evaluation is conducted by both comparing the es-

timated acceleration ratios over Woodcock tracking (WT) (*i.e.*, the case without partitioning) and comparing the actual acceleration ratios measured by shooting random rays. Here, the acceleration ratio is defined as the ratio between the averages of the expected numbers of iterations. We did not use the computation time, as it is dependent on the optimization of the code and the computing architecture.

For the evaluation, we prepared the following 168 types of participating media, shown in Figure 4, with different spatial distributions of the extinction coefficient. Using 3D Perlin noise [Per02], we first prepared 3 types of base media (represented by $8 \times 8 \times 8$ grid-cells) by setting the frequency parameter from 0 to 2. After that, we normalized the extinction coefficient in the range $[0,1]$. Then for each of the base medium, we prepared 8 types of different variations by changing the contrast to $2^j$, $(j = 1, ..., 8)$. The contrast is defined as the ratio of the maximum difference of the extinction coefficient to the average extinction coefficient. To obtain the desired contrast, we applied a power $r$ to the normalized extinction coefficients, where an appropriate $r$ is found using a bisection method. Finally, for each of these 24 types of participating media, we multiplied the extinction coefficient by $1/8$, $1/4$, $1/2$, 1, 2, 4, 8 times of a baseline value to create 7 sets of participating media with different maximum extinction coefficients. In these 168 types of participating media, a medium with higher contrast parameter, higher maximum extinction coefficient and lower frequency parameter is more inhomogeneous, because the portion with higher extinction coefficient is coagulated in a smaller region.

First, we compare the estimated acceleration ratios in Figures 5(a) to (d). We can see that the tendency of the performance is basically $KD \geq hKD \geq OC \geq UG \geq WT$. Especially, for highly inhomogeneous participating media, kd-tree based schemes (hKD and KD) performed much better.

To investigate this tendency in more detail, we show in Table 1(a) the win-loss standings between any two schemes, and in Table 2(a) a quantitative evaluation of the estimated acceleration ratios. From Table 1(a), we can see that although hKD is basically superior to UG and OC, there are about 30 to 40% cases where hKD is inferior to UG and OC. Conversely, KD is always equal or superior to UG and OC, and outperforms hKD for almost all the cases except for only 7 cases. From Table 2(a), we can see that KD performs more than 2 times better than UG or OC for highly inhomogeneous participating media.

Next, we show the actual acceleration ratios of the schemes over Woodcock tracking in Figures 5(e) to (h). These ratios were obtained by shooting random rays and measuring the actual numbers of iterations. The computation times for the space partitioning are shown in Figure 6 and are not taken into account when computing the acceleration ratios. We can see that the tendency in the results matches our estimation fairly well. The win-loss standings and the qualitative evaluation of the acceleration ratio are

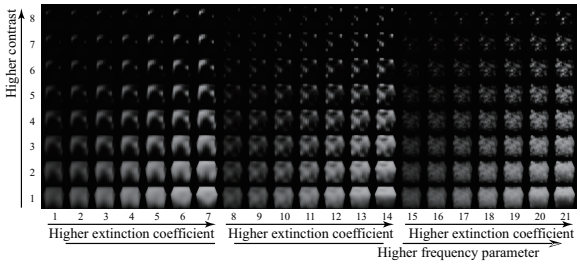


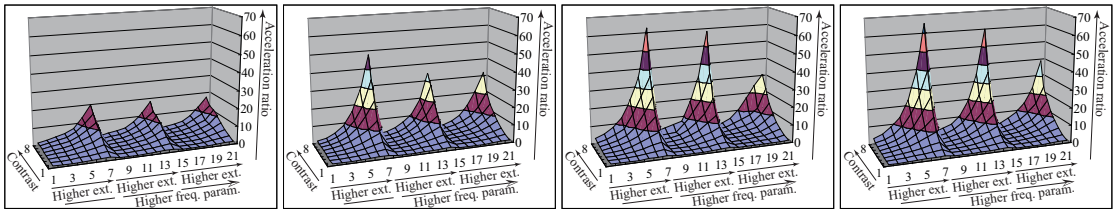

**Figure 4:** *Rendered results of the participating media used for the evaluation of the sampling efficiency. The horizontal axis corresponds to 21 sets of media with varying frequency parameter and multiplier to the extinction coefficient. The multipliers are $1/8$ in the sets 1, 8 and 15; $1/4$ in the sets 2, 9 and 16; $1/2$ in the sets 3, 10 and 17; 1 in the sets 4, 11 and 18; 2 in the sets 5, 12 and 19; 4 in the sets 6, 13 and 20; 8 in the sets 7, 14 and 21. The frequency parameters are 0, 1 and 2 in the sets 1 to 7, 8 to 14 and 15 to 21, respectively. The vertical axis shows the contrast parameter $j$.*

**Table 1:** *Win-loss standings for (a) the estimated acceleration ratios and (b) the measured acceleration ratios. For example, in (a), UG is equal or superior to WT in 168 cases and inferior in 0 case; hKD is equal or superior to OC in 102 cases and inferior in 66 cases.*

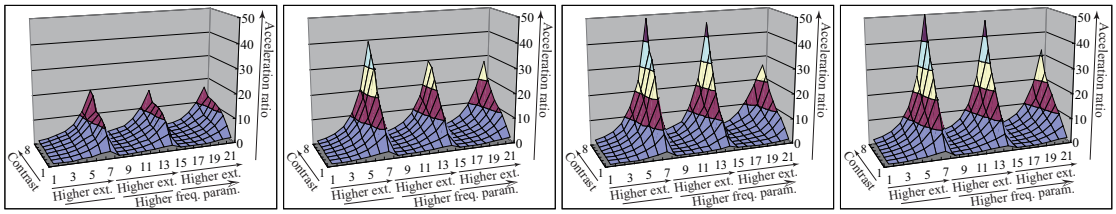
| (a) | WT | UG | OC | hKD |
|---|---|---|---|---|
| UG | 168-0 | - | - | - |
| OC | 168-0 | 168-0 | - | - |
| hKD | 168-0 | 120-48 | 102-66 | - |
| KD | 168-0 | 168-0 | 168-0 | 161-7 |
| (b) | WT | UG | OC | hKD |
| UG | 168-0 | - | - | - |
| OC | 168-0 | 165-3 | - | - |
| hKD | 168-0 | 124-44 | 102-66 | - |
| KD | 168-0 | 168-0 | 167-1 | 162-6 |

shown in Tables 1(b) and 2(b), respectively. By comparing to Tables 1(a) and 2(a), we can see that our estimation framework works fairly good (*e.g.*, the estimation error is within 10% for the average acceleration ratios). One may be curious in the fact that hKD is always superior to WT in our result, while in [YIC*10], it is shown that hKD becomes slightly inferior to WT for nearly homogeneous participating media (media with low contrast values and high frequency parameters). This is due to the difference in the settings of the media. In this paper, the frequency parameters of the media are lower, which is due to the low resolution ($8^3$) of the participating media. We would like to conduct an evaluation for higher resolution participating media in the future.

Finally, we show an evaluation using a volume data of a smoke (Figure 7). The smoke is lit by an environment light source. We calculated multiple scattering inside the smoke

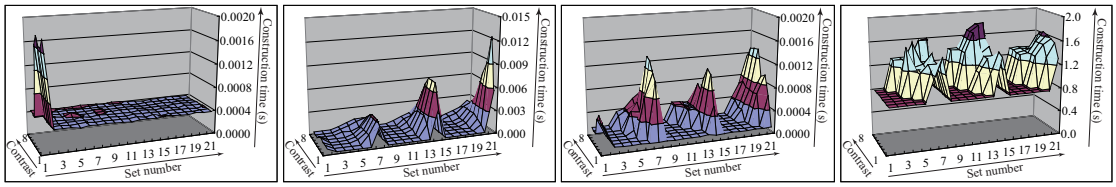



(a) UG (b) OC (c) hKD (d) KD




(e) UG (f) OC (g) hKD (h) KD

**Figure 5:** *(a) to (d) and (e) to (h): Comparisons of the estimated and measured acceleration ratios, respectively. (a) and (e): Uniform grid based scheme, (b) and (f): octree based scheme, (c) and (g): kd-tree based scheme using the heuristic approach, and (d) and (h): kd-tree based scheme presented in this paper. The vertical axis shows the acceleration ratio over Woodcock tracking. The left-right and anterior-posterior axes correspond to the horizontal and vertical axes in Figure 4, respectively.*




(a) UG (b) OC (c) hKD (d) KD

**Figure 6:** *Computation time of the space partitioning. The vertical axis shows the construction time for the spatial acceleration data structure in seconds. The left-right and anterior-posterior axes correspond to the horizontal and vertical axes in Figure 4, respectively. When using UG, the construction times are almost constant. The construction times for hKD are shorter than UG if the participating media are nearly homogeneous, and become longer otherwise. The construction times for KD are much longer than for other schemes.*

using Monte Carlo path tracing (1024 samples per pixel). We set the single scattering albedo to 0.8, and we used the isotropic phase function. When using hKD, the rendering times for Figures 7 (a) and (b) were 26.6 minutes and 44.9 minutes on a PC with an Intel Core i7 Extreme 975 CPU. We used only 1 core of the CPU and the image resolution is 480 by 960. In Figure 7 (b), using UG, OC and hKD resulted in 6.72, 8.42, 13.7 times faster rendering speeds than using WT, respectively. The resolution of the volume data is $256^3$. The construction times of UG, OC and hKD were 0.12s, 0.70s and 2.7s, respectively, and are negligible compared to the rendering times.

## 7. Conclusions and Future Work

In this paper, we have presented an estimation framework that works in the 3D space for evaluating the sampling efficiency in free path sampling, given a space partitioning. We estimated the sampling efficiency by accounting for the average of the expected numbers of iterations needed before the rays encounter 'real' scattering events. An important contribution of this paper is the analytical formulation for estimating the average of the expected numbers of iterations. Additionally, the formulation does not contain any heuristic user specified parameters. Then, we have shown that the estimation framework can be used to construct new automatic space partitioning schemes. Moreover, by using our estimation framework, we are able to estimate the difference in the sampling efficiencies among different space partitioning schemes.

For the future work, we would like to extend our framework to account for 1) the occurrence of 'real' scattering events inside the subspaces, and for 2) the case where the rays are not uniformly distributed in the space. We believe these cases could be handled by applying some weight-

**Table 2:** *Quantitative evaluation of (a) the estimated acceleration ratios and (b) the measured acceleration ratios. In the notation $\alpha(\beta,\gamma)$, $\alpha$, $\beta$ and $\gamma$ indicate the average, minimum and maximum of the acceleration ratios for the 168 cases. For example, in (a), hKD is 1.09 times superior to OC in average. In the worst case, the performance of hKD is 0.68 times to OC, and in the best case, the performance of hKD is 2.22 times to OC.*

| (a) | WT | UG | OC | hKD |
|---|---|---|---|---|
| UG | 3.80(1.0, 19.7) | - | - | - |
| OC | 5.47(1.0, 47.5) | 1.20(1.0, 2.41) | - | - |
| hKD | 6.35(1.0, 62.3) | 1.33(0.68, 3.24) | 1.09(0.68, 2.22) | - |
| KD | 6.89(1.0, 64.8) | 1.47(1.0, 3.32) | 1.21(1.0, 2.22) | 1.13(0.84, 1.66) |
| (b) | WT | UG | OC | hKD |
| UG | 4.08(1.0, 19.7) | - | - | - |
| OC | 5.60(1.0, 39.2) | 1.18(0.99, 2.30) | - | - |
| hKD | 6.09(1.0, 48.5) | 1.23(0.58, 2.61) | 1.02(0.57, 1.54) | - |
| KD | 6.64(1.0, 49.9) | 1.36(1.0, 2.65) | 1.15(0.97, 1.57) | 1.14(0.89, 1.82) |

(a) (b)

**Figure 7:** *Rendered results of a smoke under different environment light sources. The smoke (b) has 4 times higher extinction coefficient than the smoke (a).*

ing factor on the estimated values. Another interesting research direction would be to construct other space partitioning schemes based on our framework. We are seeking for a kd-tree based space partitioning scheme that utilizes some approximations during the partitioning so that the computation time needed for the partitioning is kept practical while the resulting sampling efficiency is still near optimal.

## Acknowledgments

We gratefully acknowledge helpful comments and thoughtful suggestions from the anonymous reviewers. This work was supported by JSPS Grant-in-Aid for Scientific Research (C) (23500112) and National Science Council of Taiwan (NSC99-2221-E-002-016).

## References

[BB09] BADAL A., BADANO A.: Accelerating Monte Carlo simulations of photon transport in a voxelized geometry using a massively parallel graphics processing unit. *Medical Physics 36*, 11 (2009), 4878–4880. 2

[BM03] BROWN F. B., MARTIN W. R.: Direct sampling of Monte Carlo flight paths in media with continuously varying cross-sections. In *Proc. ANS Mathematics & Computation Topical Meeting* (2003). 2

[CCW72] CARTER L., CASHWELL E., W.M.TAYLOR: Monte Carlo sampling with continuously varying cross sections along flight paths. *Nuclear Science and Engineering 48* (1972), 403–411. 2

[Cha50] CHANDRASEKHAR S.: *Radiative Transfer*. Clarendon Press, 1950. 1

[Col68] COLEMAN W.: Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems. *Nuclear Science and Engineering 32* (1968), 76–81. 2

[CPP*05] CEREZO E., PÉREZ F., PUEYO X., SERÓN F. J., SILLION F. X.: A survey on participating media rendering techniques. *The Visual Computer 21*, 5 (2005), 303–328. 2

[GJJD09] GUTIERREZ D., JENSEN H. W., JAROSZ W., DONNER C.: Scattering. In *SIGGRAPH ASIA Courses* (2009). 2

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. SIGGRAPH 98* (1998), pp. 311–320. 1, 2

[Lep07] LEPPÄNEN J.: *Development of a New Monte Carlo Reactor Physics Code*. PhD thesis, Helsinki University of Technology, 2007. 1, 2

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Proc. EGWR 96* (1996), pp. 91–100. 1

[Per02] PERLIN K.: Improving noise. In *Proc. SIGGRAPH 2002* (2002), pp. 681–682. 7

[PH89] PERLIN K., HOFFERT E. M.: Hypertexture. *Computer Graphics (Proc. SIGGRAPH 89) 23*, 3 (1989), 253–262. 2

[PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. In *Proc. EGWR 2000* (2000), pp. 11–22. 1, 2

[RSK06] RAAB M., SEIBERT D., KELLER A.: Unbiased global illumination with participating media. In *Proc. Monte Carlo and Quasi-Monte Carlo Methods 2006* (2006), pp. 591–605. 1, 2, 3

[SKTM11] SZIRMAY-KALOS L., TOTH B., MAGDIC M.: Free path sampling in high resolution inhomogeneous participating media. *Computer Graphics Forum 30*, 1 (2011), 85–97. 2

[WMHL65] WOODCOCK E., MURPHY T., HEMMINGS P., LONGWORTH T.: Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conference on the Application of Computing Methods to Reactor Problems, ANL-7050* (1965), pp. 557–579. 1, 2

[YIC*10] YUE Y., IWASAKI K., CHEN B.-Y., DOBASHI Y., NISHITA T.: Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2010) 29*, 5 (2010), 177:1–177:7. 2, 3, 6, 7